High Level Definition (HLD)

(Spam Detection)

Deep Learning Technology (NLP)

Revision Number: 1.0

Last date of revision: 18/12/2022

Document Version Control

Date Issued	Version	Description	Auth or
18/12/2022	1	Initial HLD — V1.0	Bhavya Shah

Contents

Abstract

Email spam has become a serious issue in recent years, and as the number of internet
users grows, so does the number of spam emails. They are being used for unlawful and
immoral activities, such as phishing and fraud. Sending harmful links via spam emails,
which can destroy our system as well as get access to yours. Spammers can easily create
a fake profile and email account, and in their spam emails, they appear to be a real
person. These spammers target those who are unaware of the scams.

1 Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like: o Security
  - o Reliability
  - o Maintainability
  - o Portability
  - o Reusability
  - o Application compatibility
  - o Resource utilization
  - o Serviceability

1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

1.3 Definitions

2 General Description

2.1 Product Perspective

Spam classification is the most important thing for public for its safety.

Problem statement

Email spam has become a serious issue in recent years, and as the number of internet users grows, so does the number of spam emails. They are being used for unlawful and immoral activities, such as phishing and fraud. Sending harmful links via spam emails, which can destroy our system as well as get access to yours. Spammers can easily create

a fake profile and email account, and in their spam emails, they appear to be a real person. These spammers target those who are unaware of the scams.

PROPOSED SOLUTION

Machine learning methods of recent are being used to successfully detect and filter spam emails. Some Methods are Content Based Filtering Technique, Case Base Spam Filtering Method, Heuristic or Rule Based Spam Filtering Technique or Previous Likeness Based Spam Filtering Technique.

The system should be able to easily classify between real and fake emails.
You have to build a robust antispam filter solution that should identify fake and real emails.

2.2 Tools used

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, VS Code, Matplotlib and Github, nlpn ml are used to build the whole model.

2.3 Constraints

The system should be able to easily classify between real and fake emails.

You have to build a robust antispam filter solution that should identify fake and real emails.

2.4 Assumptions

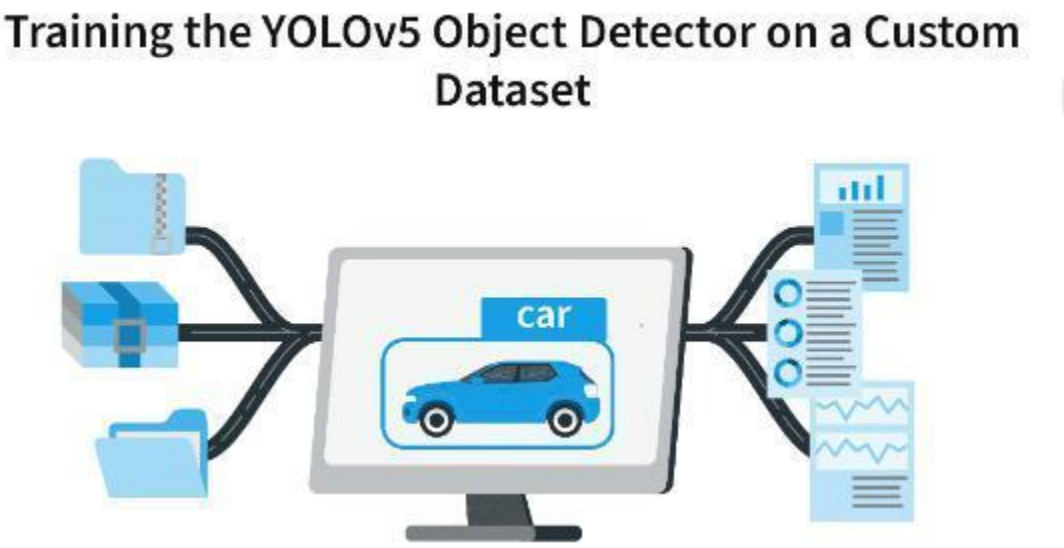
The purpose of the spam detection system is **Safe Guard people form the spam massages.**

3 Design Details

3.1 Process Flow

For identifying the different types of anomalies, we will use a machine learning base model. Below is the process flow diagram as shown below.

3.1.1 Deployment Process



3.2 Event log

The system should log every event so that the user will know what process is running internally.

Initial Step-By-Step Description:

1. The System identifies at what step logging required
2. The System should be able to log each and every system flow.
3. Developer can choose logging method. You can choose database logging/ File logging as well.
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

3.3 Error Handling

Should errors be encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

4 Performance

The purpose of the drowsiness detection system is **to aid in the prevention of accidents in passenger and commercial vehicles**.

4.1 Reusability

The code written and the components used should have the ability to be reused with no problems.

4.2 Application Compatibility

The different components for this project will be using Python as an interface between them. Each component will have its own task to perform, and it is the job of the Python to ensure proper transfer of information.

4.3 Resource Utilization

When any task is performed, it will likely use all the processing power available until that function is finished.

4.5 KPIs (Key Performance Indicators)

1. Key indicators displaying a summary of the anomaly detection store sales
2. We can train the dataset.
3. Create the artifact files.
4. Store all the logs.

5 Conclusion

The system will detect the early symptoms of drowsiness before the driver has fully lost all attentiveness and warn the driver that they are no longer capable of operating the vehicle safely.

6 References

1. Google.com